IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

NEC CORPORATION, Plaintiff, v. PELOTON INTERACTIVE, INC., Defendant.	)))))))))))	C.A. No. 22-987 (CJB) [redacted]

PLAINTIFF NEC'S LETTER BRIEF IN SUPPORT OF ITS MOTION TO STRIKE (1) EXPERT REPORT OF MR. ALEX ZAMBELLI REGARDING MICROSOFT'S SMOOTH STREAMING PRODUCT AND (2) PORTIONS OF REPORT OF DR. KEVIN ALMEROTH REGARDING INVALIDITY OF U.S. PATENT NOS. 8,752,101 AND 8,909,809

OF COUNSEL:

Robert L. Maier
Jennifer C. Tempesta
Nick Palmieri
Michael E. Knierim
BAKER BOTTS L.L.P.
30 Rockefeller Plaza
New York, NY 10112
Telephone: (212) 408-2500
Fax: (212) 408-2501
E-mail: Robert.Maier@bakerbotts.com
E-mail: Jennifer.Tempesta@bakerbotts.com
E-mail: Nick.Palmieri@bakerbotts.com
E-mail: Michael.Knierim@bakerbotts.com

Kenneth L. Dorsney (#3726)
Cortlan S. Hitch (#6720)
MORRIS JAMES LLP
3205 Avenue North Boulevard, Suite 100
Wilmington, DE 19803
(302) 888-6800
kdorsney@morrisjames.com
chitch@morrisjames.com

Attorneys for Plaintiff NEC Corporation

Dated: November 19, 2025

Dear Judge Burke:

NEC moves to strike portions of Peloton's opening expert reports: (1) the Expert Report of Mr. Alex Zambelli Regarding Microsoft's Smooth Streaming Product ("Zambelli Report"), Ex. 1, and (2) portions of the Expert Report of Dr. Kevin Almeroth regarding invalidity of U.S. Patent Nos. 8,752,101 and 8,909,809 ("Almeroth Report"), Ex. 2. The Zambelli Report and portions of the Almeroth Report improperly discuss new alleged prior art, i.e., "Smooth Streaming" and "SureStream." In addition, other portions of the Almeroth Report should be stricken for relying on invalidity theories not previously disclosed.

1. Applicable Case Law

Courts in the Third Circuit follow the multi-factor test under *Meyers v. Pennypack Woods HOA*, 559 F.2d 894 (3d Cir. 1977) to evaluate whether to exclude expert testimony based on untimely disclosure. *See Chervon (HK) Limited v. One World Technologies, Inc.*, No. 19cv1293, D.I. 472, at 15 (D. Del. Jan. 14, 2025). In particular, expert opinions relying on theories or evidence not timely disclosed are prejudicial and should be stricken, *Stodge Inc. d/b/a Postscript v. Attentive Mobile Inc.*, No. 23cv87, D.I. 657, 658 (D. Del. June 2, 2025) (Burke, J.), even if the nonmovant asserts their importance. *See Chervon (HK)*, D.I. 472, at 15. The same *Pennypack* factor analyses, as detailed in NEC's pending motion to strike, D.I. 301 ("Prior Motion"), apply here as well.

2. NEC Is Prejudiced by Peloton's Reliance on New Prior Art Disclosed at the Last Minute

As explained in NEC's Prior Motion, four days before the end of fact discovery, Peloton in its Final Invalidity Contentions ("FICs"), D.I. 301, Ex. A, presented for the first time entirely new invalidity theories for the asserted '101 and '809 Patents based on "Microsoft IIS Smooth Streaming" ("Smooth Streaming") and RealNetworks systems including SureStream ("SureStream"), which were not previously mentioned anywhere in Peloton's Initial Invalidity Contentions ("IICs"). The Zambelli and Almeroth Reports only confirm the prejudice to NEC, as explained in the Prior Motion.

The Zambelli Report purports to "provide a summary of how the Smooth Streaming technology and related services were developed at Microsoft and the technical details of Smooth Streaming design." Ex. 1 at 1. Therefore, it should be stricken in its entirety for the same reason as stated in NEC's Prior Motion, based on Peloton's belated disclosure of Smooth Streaming. Indeed, the Zambelli Report confirms the merit of NEC's Prior Motion. For example, Mr. Zambelli purports to "enumerate ***all*** the materials … relied upon or considered in [his attached] Appendix" *id.* (emphasis added), and every single reference in the Appendix is public material available online. *Id.* at 36-42. As NEC explained in its Prior Motion, there was no excuse for Peloton to have waited until the end of fact discovery to disclose Smooth Streaming as alleged prior art based on public information. D.I. 301 at 3; *see Oasis Tooling, Inc. v. GlobalFoundries U.S., Inc.*, No. 1:22-cv-00312, D.I. 428 (D. Del. Mar. 4, 2024) (Burke, J.) (granting motion to strike when plaintiff "disclosed a theory based solely on publicly-available information approximately 17 months after filing suit, … after the deadline for serving written discovery, … and four days prior to the close of fact discovery" as "untimely pursuant to Rule 26" (cleaned up)).

Similarly, Sections XII.A-B, XII.I-N, XIV.H-K, App'x D Sections I.A, I.B, II.F, II.G, of

the Almeroth Report should be stricken for the same reason as stated in NEC's Prior Motion, D.I. 301, based on Peloton's belated disclosure of the proffered Smooth Streaming and SureStream prior art. Again, these sections confirm the merit of NEC's Prior Motion by demonstrating concrete examples of prejudice to NEC, as NEC previously explained. *Id.* at 2-3. For example, in Sections XII.I and XII.L, Dr. Almeroth offers opinions regarding motivations to combine Smooth Streaming and SureStream with various other references. These motivations to combine, however, depend on many factors specific to the combination, including the particular transport technology used (e.g., RTP, HTTP, TCP), the encoding schema, any limitations of the components of the content delivery system, and the bit rate change algorithms' structure. The PTAB confirmed as much during the IPR, stating that there would not be motivation to combine references that are deployed in certain different technological contexts. Ex. 3 at 54; Ex. 4 at 47. Yet, NEC here was deprived of the ability to conduct discovery on the Smooth Streaming and SureStream systems that would have been informative to a motivation to combine analysis. It is well established that "whether a person skilled in the art would have been motivated to combine" is a "fact-intensive inquiry," and NEC was deprived of the opportunity to discover facts that would be relevant here. *Apple Inc. v. Andrea Elecs. Corp.*, 949 F.3d 697, 710 (Fed. Cir. 2020); *see also MR Techs., GMBH v. W. Digital Techs., Inc.*, 2024 WL 3082255, at *8 (C.D. Cal. Apr. 1, 2024). Peloton argued that NEC should have been able to conduct discovery by guessing what Peloton's invalidity theories might be based on its two subpoenas,[1] even though, as explained, the discovery would depend on the specific theories or ***combinations*** asserted, which were previously unknown to NEC.

Accordingly, the Zambelli Report in its entirety, and sections of the Almeroth report relying on the Zambelli Report, Smooth Streaming, or SureStream should be stricken for the same reason as explained in the Prior Motion.

3. The Almeroth Report Relies on Additional Previously Undisclosed Theories

a. The Almeroth Report Relies on Theories Based on [redacted] not Disclosed in Peloton's FICs

Even to the extent Peloton's FICs were proper (they are not), Sections XII.L-N, XIV.I, XIV.K, App'x D Sections I.B, II.G, of the Almeroth Report rely on SureStream-related theories not disclosed in Peloton's FICs. Specifically, Dr. Almeroth's testimony relies on a source code file [redacted] and the functions [redacted] to allege invalidity of both asserted patents. *See* Ex. 2 ¶¶ 256, 271-274, 281-283, 287-289, 296-298, 306-308, 1219-1221, 1227-1229, 1237-1239, 1243-1245, 1251-1253, 1257-1259, and 1263-1265 ("[redacted] Paragraphs"). However, those files and functions appear nowhere in the FICs. Instead, the FICs point to entirely ***different*** files not relied on by Dr. Almeroth (and no functions at all). *See* D.I. 301 Ex. A2 at 29, Ex. A6 at 51. This is not simply about relying on a different portion of the same evidence—the theory of invalidity is entirely different from what was included in the FICs because it relies on an entirely different codebase ([redacted]) that NEC had no notice of until it first appeared in Dr. Almeroth's expert report. These sections should be stricken for this additional reason. *Stodge*, No. 23cv87, D.I. 658 (Burke, J.) (granting motion to strike where "[t]here is no

[1] *See* D.I. 307 at 3. Indeed, not even Microsoft itself could understand what products the subpoena sought to discover. D.I. 318, Ex. A. And certainly, NEC is entitled to the notice that contentions are designed to provide—NEC should not be forced to guess at Peloton's invalidity theories.

question that [the expert opinions] were not included in Plaintiff's final infringement contentions" before the expert report). As explained above, NEC is prejudiced because it was unable to conduct fact discovery to investigate this new theory, including on the motivation to combine, which is highly dependent on the specifics of the prior art combination being asserted. Accordingly, the [redacted] Paragraphs should be stricken.

b. The Almeroth Report Also Presents Previously Undisclosed §112 Arguments and Previously Waived Claim Construction Arguments

Further, Section XI of the Almeroth Report should be stricken because it advances §112 invalidity theories previously undisclosed in the FICs. Dr. Almeroth argues that the '101 Patent is invalid under §112 for written description issues because it "does not describe how to determine code rate once reproduction has started" and "determines reproduction start time only once for the content, and not for each content data" and that "available reproduction time" and "remaining time" cannot be the same. *E.g.* Ex. 2 ¶¶ 146 (§XI.A), 151 (§XI.B), 153 (§XI.C).

Peloton disclosed none of these theories in its FICs. In Section X.1 of the FICs, Peloton merely recites certain claim language and boilerplate assertions that the claims "lacks sufficient written description and enablement." There was not a word in the FICs on the theories presented by Dr. Almeroth in Section XI of his report. D.I. 301, Ex. A at 128-130.

Moreover, Peloton specifically waived these arguments. Dr. Almeroth's §112 arguments in §§XI.A and XI.B of his report are the precise arguments Peloton and Dr. Almeroth made during claim construction for the term "reproduction start time set as time at which the reception device starts reproduction of the content," where Peloton argued that "the content has one start time" and "the code rate is updated only before the reproduction begins and not after." D.I. 137 at 4, 7. Peloton subsequently expressly abandoned these arguments and agreed to NEC's proposed construction, giving the term its "plain and ordinary meaning." D.I. 236. Dr. Almeroth is now merely resurrecting Peloton's abandoned claim construction argument (even though it was not maintained in Peloton's FICs). Notably, another court in this District previously admonished Dr. Almeroth in a similar circumstance, where he "offered opinions based on his claim constructions of terms that would otherwise be afforded their plain meaning." *D&M Holdings, Inc., et al. v. Sonos, Inc.*, No. 16cv141, D.I. 282, at 2-3 (D.Del. Feb. 6, 2018) (noting that "Dr. Almeroth cannot do claim construction" and striking Dr. Almeroth's improper construction).

4. Conclusion

Accordingly, NEC respectfully requests that (1) the Zambelli Report be stricken in its entirety (and consequently the following paragraphs of the Almeroth Report: ¶¶ 107, 159-167, 174-175, 179-180, 186-187, 193-194, 199-201, 203, 209-210, 215-218, 225-227, 229, 236-239, 245-246, 249-252, 563, 569, 575, 581, 1119, 1123-1124, 1129-1130, 1134-1135, 1139-1142, 1145-1148, 1152, 1155-1158, 1162, 1167-1171, 1174-1179, 1184-1185, 1187-1190, 1192-1197, 1200-1205 and App'x D ¶¶ 25, 503) and Sections XII.A-B, XII.I-N, XIV.H-K, App'x D Sections I.A, I.B, II.F, II.G, of the Almeroth Report be stricken for relying on Smooth Streaming and SureStream; and (2) (a) ¶¶ 256, 271-274, 281-283, 287-289, 296-298, 306-308, 1219-1221, 1227-1229, 1237-1239, 1243-1245, 1251-1253, 1257-1259, and 1263-1265 of the Almeroth Report and (b) Section XI of the Almeroth Report be stricken.

Respectfully,

/s/ Cortlan S. Hitch
Cortlan S. Hitch (#3726)

cc: All Counsel of Record (via electronic mail)

EXHIBIT 1

[redacted]

IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

NEC CORPORATION,	)	
	)	
Plaintiff,	)	
	)	C.A. No. 22-987-CJB
v.	)	
	)	
PELOTON INTERACTIVE, INC.,	)	
	)	
Defendant.	)	

EXPERT REPORT OF MR. ALEX ZAMBELLI REGARDING MICROSOFT'S SMOOTH STREAMING PRODUCT

TABLE OF CONTENTS

I. INTRODUCTION

1. My name is Alex Zambelli, and I have been retained by Peloton Interactive, Inc. ("Peloton") as an expert witness in the above captioned proceeding to provide an overview of the design and operation of Microsoft's Smooth Streaming technology and related products and services.

2. I am being compensated on a per-hour basis for my time spent on conducting background research, factual investigation and preparing this report. I am being compensated at the rate of $500 per hour. My compensation does not depend upon the outcome of this matter or the content of this report and any opinions that I provide.

II. SCOPE OF WORK

3. In this report, I summarize my qualifications and my employment history. I then provide a summary of how the Smooth Streaming technology and related services were developed at Microsoft and the technical details of Smooth Streaming design.

4. My report is based on my years of education, research and work experience – my work at Microsoft in particular. I rely upon documents and blog posts that I authored while I worked at Microsoft, documents I reviewed during my time at Microsoft, PowerPoint slides that I prepared and presented as part of my work at Microsoft, videos of presentations that I made as part of my work at Microsoft, documents related to Smooth Streaming that I was aware of based on my work at Microsoft, and news articles that describe relevant events and product releases that occurred during my time at Microsoft. I enumerate all the materials that I relied upon or considered in Appendix entitled MATERIALS CONSIDERED.

III. QUALIFICATIONS

A. Introduction

5. I have worked in the streaming media industry for more than two decades, spanning the formative years of online video through to today's era of mainstream adoption. My career has centered on the design, development, and management of technologies that make large-scale video streaming possible - from codecs and transport protocols to end-to-end content delivery workflows.

B. Undergraduate Education and Internship (1998-2002)

6. From 1998 to 2002 I attended Florida Institute of Technology where I pursued an undergraduate degree in Computer Science. My introduction to digital media technologies came in 2001 when I completed a summer internship at Microsoft Corporation, working as a Software Test Engineer on the Windows Media Player team at Microsoft's main campus in Redmond, Washington. After returning to Florida Institute of Technology and resuming my undergraduate studies I earned my Bachelor of Science degree in Computer Science in May 2002, graduating with honors. My academic training provided a strong foundation in computer science, mathematics, and systems engineering, which I have applied throughout my career in digital media and streaming technologies.

C. Microsoft (2002-2012)

7. My professional career began in 2002 when I was hired full-time by Microsoft and rejoined the Windows Media Player team as a Software Test Engineer. During my time on the team I was mainly responsible for testing media playback features of the product, which included early streaming capabilities. In 2005 I changed teams and joined the Core Media Processing Technology team as a Software Development Engineer in Test responsible for testing Microsoft's VC-1 (WMV9) video codec.

8. In 2007 I transitioned to a Technical Evangelist role on the Core Media Processing Technology team. In my new role I worked to familiarize video industry professionals with Microsoft's video encoding technology, and assisted customers with implementing and deploying products such as the VC-1 Encoder SDK. It was in this capacity that in early 2008 I first became involved with adaptive streaming technologies when my team was tasked with developing an adaptive bitrate streaming solution for NBC's 2008 Beijing Olympics website.

9. When in 2008 I moved to Microsoft's Developer & Platform Evangelism group the scope of my role as a Technical Evangelist (and later Senior Technical Evangelist) expanded beyond VC-1 codec to also include supporting other Microsoft media technologies such as Smooth Streaming, IIS Media Services (later Azure Media Services), Silverlight, Expression Encoder and PlayReady. As part of my expanded role I served as a solutions architect for several of the earliest large-scale live streaming events that utilized adaptive bitrate streaming, including 2009-2010 NBC Sunday Night Football, 2010 Vancouver Olympics, 2010 TNT Sports NASCAR, and 2012 London Olympics. During the same period I also started a website (alexzambelli.com) where I frequently published blog posts about Microsoft media technologies, most of which were focused on Smooth Streaming and HTTP-based adaptive bitrate streaming in general.

D. iStreamPlanet (2013-2016)

10. In 2013 I left Microsoft to join iStreamPlanet Co. as a Principal Product Manager. At iStreamPlanet I was responsible for defining requirements, designing user experiences and managing development of Aventus, a cloud-based video encoding solution. During my time with the company Aventus was used by major broadcasters such as NBC, Turner, and Fox to deliver live linear channels and large-scale events, including several

Olympics and NFL Super Bowl games. The success of Aventus was a major contributing factor to the company's acquisition by Turner Broadcasting in 2015.

E. Hulu (2016-2021)

11. In 2016 I joined Hulu as a Principal Product Manager, overseeing the company's VOD and live video platforms from ingest to playback. I led initiatives that brought 4K UHD, HDR, and high-frame-rate playback to Hulu streaming service; standardized audio loudness normalization; implemented quality-of-experience analytics; and drove adoption of advanced codecs such as HEVC. My work there combined technical innovation with a focus on consistent user experience across millions of viewing sessions.

F. Warner Bros. Discovery (2021-2023)

12. From 2021 to 2023 I served as Technical Product Manager at Warner Bros. Discovery, where I product managed VOD and live media processing platforms that supported Max (HBO Max) and Discovery+ streaming services. My contributions included defining transcoding and packaging requirements for both VOD and live video distribution, and overseeing improvements in video quality and audio quality.

G. Dolby Laboratories (2023-present)

13. In 2023 I joined Dolby Laboratories where I'm currently employed as a Senior Platform Manager for Dolby Vision encoding and OTT enablement. In this role I manage Dolby Vision's core technology components, and oversee development of specifications, tools, test content and playback test apps that enable seamless integration of Dolby's video and audio formats into modern streaming workflows.

H. Summary

14. Over the course of my career I have accumulated more than twenty years of experience in streaming media and digital video technologies. My work has spanned software testing,

software development, technical evangelism, product management and platform management, with a consistent focus on advancing video and audio technologies for large-scale consumer applications.

15. My resume listing my education, work experience, my publications, and my prior testifying experience is attached as an Exhibit to this report.

IV. EMPLOYMENT AT MICROSOFT

A. Windows Media Player

16. I joined Microsoft in 2002 as a Software Test Engineer (STE) in the Windows Media division. My early work focused on testing and validating Windows Media Player playback capabilities and related components across both Windows desktop and mobile platforms. These projects spanned versions 9 and 10 of Windows Media Player and involved detailed analysis of playback behavior, performance optimization, and codec interoperability under a variety of hardware and operating system conditions. Among features I regularly tested was Windows Media Player's ability to stream media over the Internet using packetized streaming protocols such as RTSP and MMS which were state-of-the-art streaming technology at that time.

B. Core Media Processing Technology

17. By 2006 I had moved to Microsoft's Core Media Processing Technology (CMPT), a group responsible for the design and development of Microsoft's video and audio compression technologies. My work there focused primarily on VC-1 - the standardized implementation of Microsoft's WMV9 video codec. I was at first responsible for testing hardware-accelerated VC-1 decoding in Windows Vista, and then later for testing the quality of VC-1 encoding in Windows Vista and 3rd party encoding products such as the Inlet Technologies Spinnaker. This role deepened my understanding of video compression theory, codec design, and practical applications of video compression in live streaming use cases.

18. In July 2007, while still part of Core Media Processing Technology, I transitioned into a new role as a Technical Evangelist for the VC-1 codec. At that time, technical evangelism was an established professional discipline within Microsoft - focused on

bridging engineering and the broader developer ecosystem. The goal was not to sell products, but to explain and demonstrate Microsoft technologies in a technically accurate, developer-oriented manner.

19. As a Technical Evangelist embedded in CMPT, I worked closely with both internal engineering groups and external software and hardware vendors to help them implement and optimize support for VC-1. My earliest evangelism efforts centered on codec integration, encoding best practices, and workflow design for professional video production and distribution systems.

20. In January 2008 NBC Universal announced it was partnering with Microsoft to expand NBC's online coverage of the upcoming Summer Olympic Games in Beijing by offering live and on-demand video of most Olympic events on NBC's Olympics website. As part of the joint project NBC asked Microsoft to create a new streaming solution for the NBC Olympics website that could provide a more reliable, higher-quality online video experience than the existing Windows Media streaming architecture. The task of developing this new streaming solution, an implementation of adaptive bitrate streaming (ABR), was given to Core Media Processing Technology – which is how I became involved with the project. [25]

21. My initial role in the project focused on coordinating with NBC's encoding software vendors Digital Rapids and Anystream to integrate VC-1 encoding into their products for the upcoming Olympics. As the project evolved, I also authored encoding specifications and guidelines for NBC's content preparation, defining optimal parameters for multi-bitrate encoding and segment generation to support adaptive playback.

22. The resulting technology - combining multi-bitrate video encoding, segmented content delivery over HTTP, and client-side bitrate adaptation - represented Microsoft's first successful implementation of HTTP-based adaptive bitrate streaming. It was the technical foundation of what later would be developed into an officially supported product known as Smooth Streaming, which I describe in more detail in section V.

C. Developer & Platform Evangelism

23. In March 2008 I joined Microsoft's Developer & Platform Evangelism (DPE) group where I remained for the next 5 years, serving as a Technical Evangelist and later as a Senior Technical Evangelist for Microsoft media technologies. Since my new team was responsible for managing Microsoft's participation in the NBC Olympics project at that time I continued to be involved in the project for the entirety of its duration. [5]

24. After the Olympics, the focus of my role expanded beyond the VC-1 codec to encompass a wider range of technologies within the Microsoft media ecosystem. These included Smooth Streaming, IIS Media Services (later Azure Media Services), Silverlight, Expression Encoder, and PlayReady DRM. I acted as a bridge between Microsoft's engineering teams, external media companies, encoding vendors, content delivery networks (CDNs), and broadcasters, ensuring that Microsoft's media technologies could be successfully integrated into large-scale streaming workflows.

25. My work in DPE was both technical and outward-facing. I collaborated with product teams inside Microsoft - including those developing Silverlight, IIS Media Services, Expression Encoder, and PlayReady - to ensure alignment across the company's streaming ecosystem. I also worked extensively with external partners and customers such as NBC, CTV, NRK, Turner Broadcasting, NVIDIA, iStreamPlanet, Inlet

Technologies, Elemental Technologies, Akamai, Digital Rapids, Rhozet and many others to build and deliver streaming solutions and products.

26. Between 2008 and 2012, I was involved in several landmark streaming projects as a solutions architect and/or consultant:

a) Sunday Night Football Extra (2009–2010): Live streaming of NFL broadcasts for NBC (USA).

b) 2010 Vancouver Winter Olympics: Live and on-demand streaming for NBC (USA), CTV (Canada), and NRK (Norway).

c) TNT NASCAR RaceBuddy 3D (2010): A collaboration with NVIDIA and Turner Sports that demonstrated the viability of live Smooth Streaming in 3D.

d) 2012 London Summer Olympics: Live and on-demand streaming for France TV (France), RTVE (Spain) and Terra (Mexico).

27. In addition to my work on high-profile streaming projects, during this time I also produced a series of technical blog posts and public documents on a variety of topics related to Microsoft media technologies, including an influential 2009 technical paper titled "IIS Smooth Streaming Technical Overview," published by Microsoft and later frequently cited in academic research papers. I frequently spoke about Smooth Streaming and Silverlight at conferences such as Streaming Media West and Microsoft's MIX, and gave technology demos and presentations at broadcast/streaming industry tradeshows such as NAB (Las Vegas) and IBC (Amsterdam). My professional blog, alexzambelli.com, became a respected source of technical information within the streaming media community, focusing on adaptive streaming, encoding practices, and the evolving architecture of internet video delivery. [13][14][15]

V. EVOLUTION OF SMOOTH STREAMING AT MICROSOFT

A. State of Microsoft Streaming Prior to 2008

28. Prior to 2008, Microsoft's solution for streaming video over the Internet relied on the Windows Media suite of software products such as Windows Media Services, Windows Media Player and Windows Media Encoder, as well as a rich ecosystem of 3rd party media encoder products that supported Windows Media codecs and protocols. Windows Media streaming architecture at that time was based on packetized protocols such as RTSP (Real-Time Streaming Protocol), where the server and player maintained a live, continuous connection: the server "pushed" data packets to the client in real time, and the client attempted to keep up. Network congestion could easily interrupt playback, and early attempts to introduce dynamic bitrate switching (such as Windows Media Intelligent Streaming) failed to adequately address the problem. In practice, this traditional streaming architecture based on packetized protocols worked well for internal enterprise networks and small-scale Internet webcasts, but it struggled to reliably deliver high-quality experiences to large Internet audiences due to the diversity and unpredictability of public networks. [6]

B. Microsoft Adaptive Streaming Prototype: 2008 NBC Olympics

29. In January 2008 at the annual Consumer Electronics Show in Las Vegas, Microsoft and NBC announced a partnership to expand NBC's online coverage of the upcoming Summer Olympic Games in Beijing by offering live and on-demand video of most Olympic events on NBC's Olympics website, thereby enabling U.S. viewers to watch most Olympics events from their PCs. The arrangement marked one of the first serious pushes by a major broadcaster to combine broadcast TV rights with a robust online streaming presence. [25]

30. In addition to helping NBC build a modern interactive website based on Microsoft's new Silverlight technology (Microsoft's answer to Adobe Flash), Microsoft was asked by NBC to provide an improved, more robust, higher-quality video streaming experience than what was possible with Microsoft's Windows Media architecture. HTTP-based adaptive streaming was a new approach to streaming that had started gaining attention in the streaming media industry in July 2007 when the U.S. broadcast network ABC launched their video streaming website abc.go.com using an adaptive streaming platform built by Move Networks. [1][2][3][4]

31. The new streaming architecture for NBC Olympics needed to be stateless, based on HTTP, easily scalable using existing content delivery networks (CDN) infrastructure, and capable of dynamically adjusting video quality in response to changing network conditions, ensuring a seamless playback experience even over unreliable Internet connections. The task of building this new streaming solution was assigned to Microsoft's Core Media Processing Technology team (which I was a member of at the time) because the team already held deep expertise in codecs and media processing.

32. Existing Microsoft codec technology – specifically VC-1 video codec and WMA Pro audio codec – were determined to already be suitable for encoding content in a way that would enable seamless client-side switching between quality levels. In the case of VC-1 video, seamless quality switching could be enabled by encoding the same video content at multiple resolutions and bitrates while configuring all encoder instances to use the same fixed interval (typically 2-5 seconds) between intra-coded frames (also known as keyframes or I-frames). Inserting keyframes at exactly the same points in the content timeline across all encoded renditions made it possible for the encoded streams to then be

split at keyframe boundaries into short segments (known as chunks), which in turn made it possible during playback for adaptive streaming clients to switch between encoded video renditions at any chunk boundary without the risk of repeating or skipping some part of the content timeline. [8][15]

33. CMPT team thus became responsible for developing a tool that could:

a) take as input multiple full-length VC-1-compressed video files representing the same content encoded at different quality levels (different bitrates and/or resolutions) while configured with the same fixed-length keyframe interval,

b) split each full-length video file into short chunks so that every new chunk file began with a keyframe and could therefore be played independently of any other chunk in the sequence,

c) create an index (manifest) file describing video and audio streams of the media content presentation, their different renditions and quality levels, their chunk sequences, and various timing information necessary for synchronized playback, and

d) package all the created file chunks into a single archive file that could then be easily uploaded to a web server and unpacked back into individual chunk files, ready to be served to clients. [8][15]

34. CMPT team was also responsible for developing the core adaptive bitrate (ABR) client module that could be used by a Silverlight-based media player application to:

a) parse and evaluate the media content presentation's XML-formatted index file (also known as a manifest),

b) build a logical map of the media presentation's video and audio streams and their multi-bitrate renditions (quality levels), and

c) make intelligent decisions about which media chunk to download next by comparing the chunk's available bitrates (quality levels) to a multitude of optimization factors such as chunk download speeds, estimated network bandwidth, and player buffer fullness.

35. While CMPT was responsible for development of these core components of the ABR streaming solution, Microsoft's Developer & Platform Evangelism group was responsible for coordinating the overall project with NBC Sports, internal Microsoft product teams such as Silverlight and MSN, and external vendors such as Digital Rapids, Anystream, Delta Tre, Limelight Networks, and others. My own responsibilities during the project included working with NBC's media encoding software vendors Anystream and Digital Rapids to integrate support for VC-1 and WMA Pro codecs and ABR-ready encoding into their products in time for the Olympics, and authoring encoding specifications to guide NBC's transcoding workflows for both legacy Windows Media streaming and new HTTP-based adaptive bitrate streaming. [8][15][26]

36. As work on the project progressed during spring and summer of 2008, Microsoft filed multiple patent applications based on inventions made while developing the ABR streaming solution for NBC Olympics. Published patents stemming from the project included:

a) Sanjeev Mehrotra, Kishore Kotteri, Bharath Siravara, Thomas W. Holcomb, Hui Gao, and Serge Smirnov. *"Optimized Client Side Rate Control and Indexed File Layout for Streaming Media."* US Patent 8,379,851, issued 19 February 2013.

b) Wenbo Zhang, Serge Smirnov, Kishore Kotteri, Gurpratap Virdi, Eldar Musayev, and Florin Folta. *"Media Streaming Using an Index File."* US Patent 7,925,774, issued 12 April 2011.

c) Gurpratap Virdi, Andres Vega-Garcia, Serge Smirnov, Wenbo Zhang, and Eldar Musayev. *"Stream Selection for Enhanced Media Streaming."* US Patent 7,949,775, issued 24 May 2011.

d) Gurpratap Virdi, Andres Vega-Garcia, Wenbo Zhang, and Eldar Musayev. *"Media Streaming with Enhanced Seek Operation."* US Patent 8,370,887, issued 5 February 2013.

e) Thomas W. Holcomb, Sanjeev Mehrotra, Serge Smirnov, and Bharath Siravara. *"Encoding Streaming Media as a High Bit Rate Layer, a Low Bit Rate Layer, and One or More Intermediate Bit Rate Layers."* US Patent 8,325,800, issued 4 December 2012.

37. All inventors listed on those patents were members of the Core Media Processing Technology team at the time the applications were filed, and many of them – Kishore Kotteri, Bharath Siravara, Thomas Holcomb, Serge Smirnov, Gurpratap Virdi – I personally worked with and was aware of their direct involvement in design and development of the adaptive streaming solution built for the NBC Olympics website. The deployed system's functionality and behavior appeared consistent with the foundational concepts and design principles described in those patents.

38. The adaptive streaming solution built by Microsoft was deployed to U.S.-based production servers and integrated into NBC's Silverlight-based video player in time for the start of the Beijing Olympic Games on August 8th, 2008. Over the next several weeks

the system delivered thousands of hours of on-demand video using HTTP-based adaptive streaming, covering nearly every Olympic event. Over the course of the games NBCOlympics.com attracted over 40 million unique visitors and served more than 50 million video streams. Despite this unprecedented scale, the streaming platform remained stable and performed reliably. [27]

39. From a technical standpoint, the project demonstrated that HTTP-based adaptive streaming was not only viable but also capable of supporting one of the largest streaming events in history up to that point. The experience validated many of our architectural choices – multi-bitrate encoding, stream segmentation, and client-driven bitrate adaptation – and set the stage for productization.

C. Productization of Microsoft Adaptive Streaming

40. Following the completion of the 2008 Beijing Olympics, Microsoft began the process of formalizing the adaptive streaming solution developed for NBC into a supported software product. Responsibility for developing the new product was assigned to the IIS Media Services team which had recently been formed from members of the former Windows Media Services product team after Microsoft decided to shift its streaming efforts from specialized media servers toward HTTP delivery on standard web infrastructure. IIS Media Services became the new home for Microsoft's server-side media technologies, and adaptive bitrate streaming became its flagship product feature. The plan was to integrate adaptive streaming support into IIS Media Services, a free downloadable extension for Internet Information Services (IIS), Microsoft's web server software. This would allow Windows Server and IIS customers to deliver adaptive bitrate content using the same infrastructure they used for conventional web hosting. [8]

41. Sometime during the latter part of 2008 Microsoft's adaptive streaming technology was given an official product name – Smooth Streaming. During its product lifetime it was also known as IIS7 Smooth Streaming, IIS Smooth Streaming and Microsoft Smooth Streaming (MSS). The IIS Media Services team was primarily responsible for developing the server and client components of Smooth Streaming while coordinating with other Microsoft groups responsible for complementary technologies, including Zune Video, Expression Encoder, Silverlight, and PlayReady DRM.

42. Silverlight was Microsoft's web application development framework that launched in 2007 as Microsoft's competitive answer to Adobe Flash, and during the 2007-2012 period it was heavily promoted by Microsoft to developers building rich media applications for web browsers. Accordingly, both the Olympics prototype as well as the earliest productized Smooth Streaming implementations of client-side adaptive bitrate streaming functionality were built to support Silverlight web applications. The first implementation of Smooth Streaming client to be made publicly available to developers for use in their Silverlight applications was released in December 2008 as part of Microsoft Expression Encoder 2 SP1, which incidentally also was the first Microsoft product release which included support for encoding multi-bitrate videos in Smooth Streaming format.

43. As part of the productization process and drawing on lessons learned from the Olympics project, Microsoft made some optimizations to the original technical design of adaptive streaming. It defined a new chunk file format based on ISO Base Media File Format (also known as MP4) and a new manifest format (based again on XML), replacing the chunk and manifest formats used during the Olympics. Another key departure from the original

design was the replacement of physical file chunks with virtual chunks. The new chunk format utilized a feature of the MP4 standard known as "movie fragments" which allowed a video or audio stream to be stored within a single MP4 file as a sequence of clearly delineated and indexed video/audio fragments. This solved the problem of having to manage thousands of files on disk for each content title, while still allowing the server to serve individual chunks to clients. When a Smooth Streaming client requested a time-based chunk from the server, the server would simply look up the location of the appropriate movie fragment box within the MP4 file, extract it and send it to the client as a virtual chunk. [8][9][15]

D. Smooth Streaming: From Product Announcement to Launch

44. Smooth Streaming was first announced to the public in October 2008 at the Digital Hollywood conference in Los Angeles. Microsoft announced the technology jointly with Akamai Technologies, one of the largest global content delivery networks at the time. According to Microsoft's official press release dated October 28, 2008, the companies demonstrated Smooth Streaming as an enhancement to Silverlight's media capabilities, enabling adaptive delivery of HD content over HTTP. Trade publications such as Streaming Media also reported on the announcement, describing Smooth Streaming as a new method for adaptive bitrate delivery integrated with Microsoft's IIS platform and Silverlight player technology. [28][29]

45. Shortly after the Digital Hollywood announcement Microsoft and Akamai launched SmoothHD.com, a demo website designed to showcase Smooth Streaming's capabilities. The site delivered high-definition demo content encoded at multiple bitrates and resolutions using the Smooth Streaming architecture. SmoothHD.com served as both a marketing tool and a real-world test platform for Microsoft's development of Smooth

Streaming's server technology component. I participated in the launch of SmoothHD.com by personally encoding most of the demo content available on the site at launch. [7]

46. In February 2009, Microsoft released the first Smooth Streaming beta, which provided support for video-on-demand (VOD) content delivery. This version was made available as a free download from Microsoft's IIS website and included technical documentation and developer guidance. The February beta was limited to VOD functionality, but it marked the first time developers and streaming industry professionals could directly experiment with adaptive streaming technology outside of Microsoft's own infrastructure. [10][11]

47. One month later, in March 2009, Microsoft released a beta version of Smooth Streaming with live streaming support as part of IIS Media Services 3.0 beta. This version added the ability to ingest and deliver live media streams in real-time. These early betas allowed partners and developers to begin testing Smooth Streaming with both pre-recorded and live media content while Microsoft continued developing and refining the product.

48. On April 20, 2009, at the NAB Show in Las Vegas, Microsoft announced the release of IIS Media Services 2.0, which included a general availability (non-beta) version of Smooth Streaming for VOD. [16][17][35]

49. Later that year, in October 2009, Microsoft released IIS Media Services 3.0, which added general availability (non-beta) support for live Smooth Streaming. This version was the first general availability release which combined full support for both live and on-demand adaptive streaming within IIS Media Services. [24]

50. From the moment it was tasked with development of Smooth Streaming in August/September 2008 and continuing through the rest of 2008 and 2009, the IIS Media

Services team stayed continuously focused on development and improvement of Smooth Streaming technology as it quickly became the flagship feature of IIS Media Services. I estimate there were between 10 and 30 people working full-time on IIS Media Services during that time, including software development engineers, QA engineers, product managers, marketing managers and business developers.

E. MIX09 Presentations

51. Release of IIS Media Services 3.0 beta (with Live Smooth Streaming beta support) was announced at Microsoft's MIX09 conference which took place March 18-20, 2009 in Las Vegas. MIX was an annual conference focused on web design and development, hosted by Microsoft and targeting software developers, UX designers and other members of the tech industry. Its attendance was generally in the range of several thousand people. [33]

52. MIX09 was the first industry event following the October 2008 announcement of Smooth Streaming where Smooth Streaming was discussed in technical detail in front of an audience. John Bocharov, Program Manager on Microsoft's IIS Media Services team and one of inventors credited on Microsoft's adaptive streaming patents, gave a talk titled "Delivering Media with IIS Media Services" in which he described Smooth Streaming architecture and benefits, and introduced live Smooth Streaming support. [12]

53. I gave a 75-minute talk titled "Silverlight Media End-to-End" in which, among other topics, I discussed Smooth Streaming as the newest method of media delivery to Silverlight clients. My presentation included descriptions of key principles and concepts behind Microsoft's adaptive streaming technology: multi-bitrate encoding, segmented

media, segment index manifests, stateless HTTP servers, scalable delivery, client-side bitrate selection logic. [13][14]

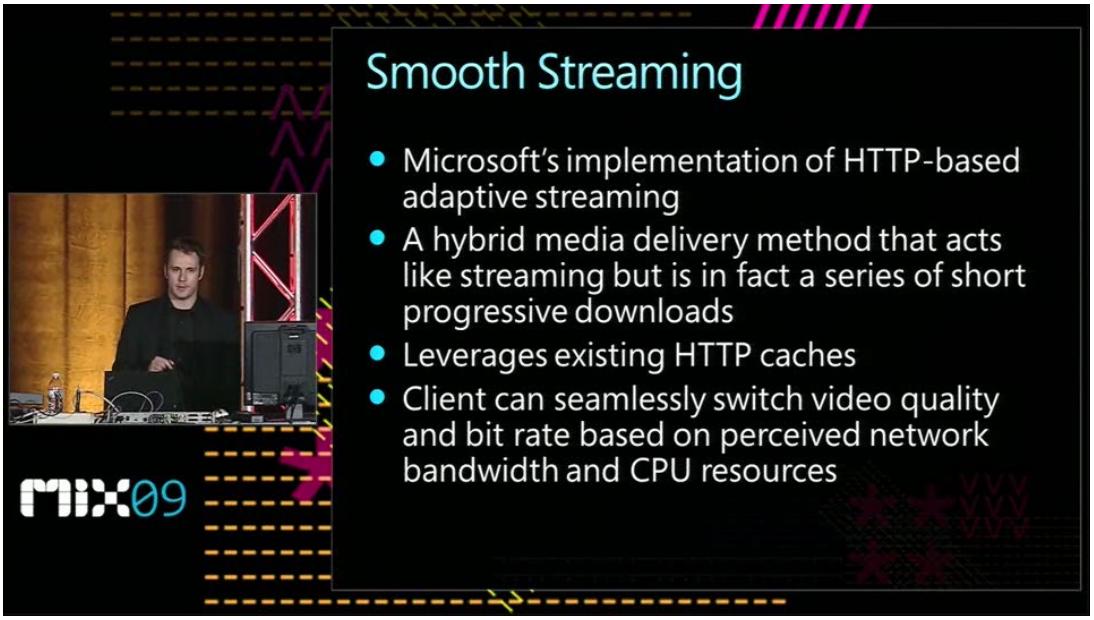

Figure 1: Screenshot from 00:30:19 of my MIX09 talk [14]

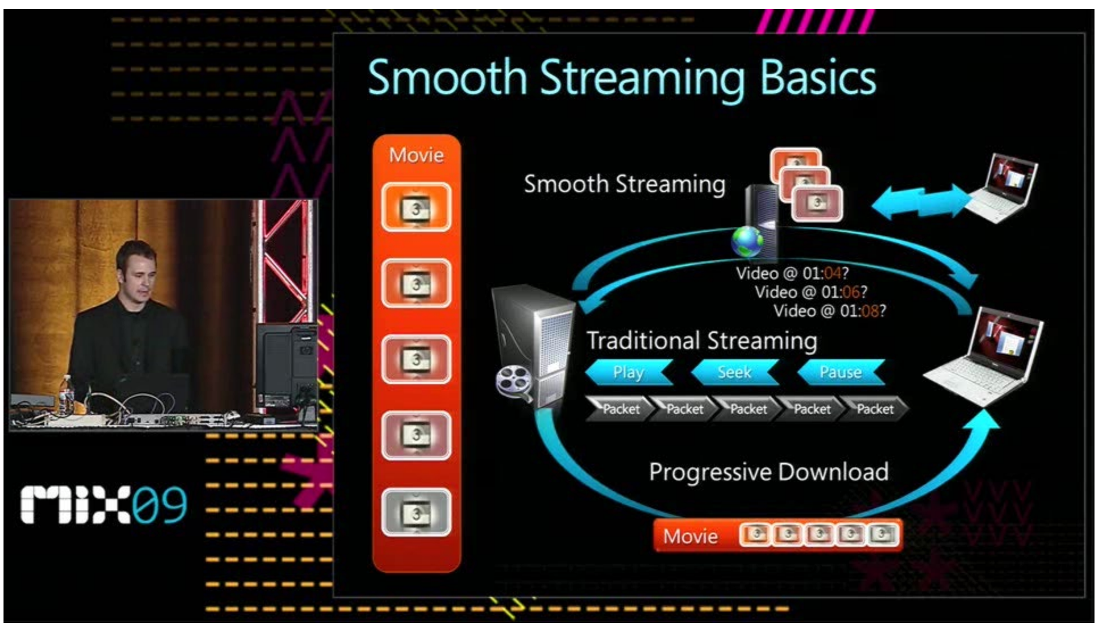

Figure 2: Screenshot from 00:32:34 of my MIX09 talk [14]

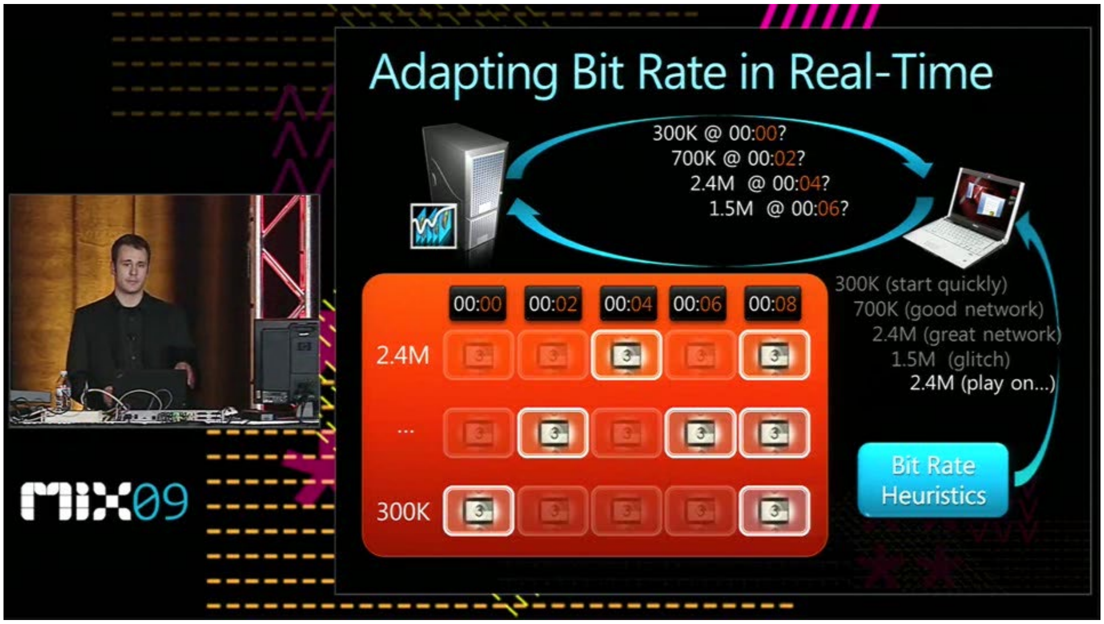

Figure 3: Screenshot from 00:35:05 of my MIX09 talk [14]

F. First Live Events in Smooth Streaming

54. While the initial releases of Smooth Streaming only supported video on-demand (VOD) delivery, after the release of IIS Media Services 3.0 beta in March 2009 much of the focus shifted to development and deployment of live Smooth Streaming which was seen as particularly valuable to broadcasters looking to expand their live sports coverage to the web. Some of the notable live events that were streamed in Smooth Streaming in 2009 included:

a) May-June 2009: Roland Garros French Open tennis tournament (France TV) [18]

b) June-July 2009: Wimbledon Championships tennis tournament (NBC Sports) [19]

c) July 2009: Tour de France cycling tour (France TV)

d) July 7, 2009: Michael Jackson Memorial (Sympatico / MSN inMusic) [20]

e) August-December 2009: NFL Sunday Night Football (NBC Sports) [22][3]

G. Smooth Streaming Patents

55. Productization of Smooth Streaming resulted in additional inventions related to adaptive streaming, for which Microsoft continued to file patent applications throughout 2008 and 2009. Published patents based on Smooth Streaming development during the productization phase included:

a) John A. Bocharov, Geqiang Zhang, Gurpratap Virdi, and Vishal Sood. *"Dynamic Fragmentation of Digital Media."* US Patent No. 8,996,547, issued 31 March 2015.

b) John A. Bocharov, Geqiang (Sam) Zhang, Jack E. Freelander, Krishna (Prakash) Duggaraju, Sudheer Sirivara, Lin Liu, Anirban Roy, Jimin Gao, Christopher G. Knowlton, and Vishal Sood. *"Delivering Cacheable Streaming Media Presentations."* US Patent No. 8,909,806, issued 9 December 2014.

c) Vishal Sood, Jack E. Freelander, Anirban Roy, Lin Liu, Geqiang (Sam) Zhang. Krishna Duggaraju, Sudheer Sirivara, John A. Bocharov. *"Smooth, stateless client media streaming."* US Patent No. 8,621,044, issued 31 December 2013.

56. Most inventors listed on those patents were members of the IIS Media Services team at the time the applications were filed.

H. Conclusion

57. By the end of 2009, Smooth Streaming supported both live and on-demand delivery, had been adopted by multiple content providers, deployed in production, supported by multiple independent software vendors, and was fully integrated into Microsoft's media product ecosystem. From my vantage point in the Developer and Platform Evangelism group, I witnessed and contributed to this transition firsthand – supporting the productization process by educating developers, partners, streaming industry professionals, and the general public about Smooth Streaming.

VI. TECHNICAL DESCRIPTION OF SMOOTH STREAMING

A. Introduction

58. Microsoft's Smooth Streaming was an HTTP-based adaptive streaming architecture first developed between 2008 and 2009 as part of the IIS Media Services platform. Its technical design was first detailed publicly in a series of blog posts that I published to my website in February 2009, which I then combined into a single technical paper (white paper) titled "IIS Smooth Streaming Technical Overview" that was published by Microsoft on its website in March 2009. [8][9][15]

59. The formal specification for the Smooth Streaming transport protocol – describing the Smooth Streaming manifest format, fragment format, and the interaction between Smooth Streaming client and server – was first published by Microsoft in September 2009 under the name *[MS-SMTH] IIS Smooth Streaming Transport Protocol*, later renamed to *[MS-SSTR] Smooth Streaming Protocol*. The file format used to store Smooth Streaming content on disk and the methods for encrypting Smooth Streaming media for use with digital rights management (DRM) systems were fully described in the *Protected Interoperable File Format* (PIFF) specification, first published by Microsoft in September 2009. [34]

B. Design Goals and Advantages

60. Smooth Streaming was designed to address the scalability and reliability challenges of earlier Windows Media–based systems by transferring the logic of adaptive bitrate (ABR) selection from the server to the client. Rather than relying on persistent streaming sessions or proprietary protocols, Smooth Streaming used standard HTTP to request small, time-indexed file fragments from a web server. This approach allowed seamless integration with existing Internet infrastructure such as IIS web servers, CDNs, and

HTTP caching proxies, enabling highly scalable video delivery without specialized streaming servers.

61. Compared to previous streaming architectures, Smooth Streaming offered several measurable technical advantages:

a) Protocol simplicity: Operated entirely over HTTP, eliminating the need for custom streaming ports or NAT traversal logic.

b) Caching efficiency: Standard CDNs and HTTP proxies could cache fragments for repeated use.

c) Scalability: Stateless design allowed load balancing and horizontal scaling using standard IIS infrastructure.

d) Client control: Bitrate adaptation occurred entirely client-side, minimizing server complexity and enabling individualized performance optimization.

e) Standardization: Based on fragmented MP4, facilitating interoperability and extension to other codecs and timed text formats.

62. The architecture also reduced operational costs for content providers by allowing media delivery over standard web servers and CDNs, rather than requiring dedicated streaming servers or specialized network equipment.

C. System Overview

63. At its core, Smooth Streaming architecture consisted of four major components:

a) A media file and fragment format based on fragmented MP4 files,

b) A manifest format based on XML which described a media presentation’s component media streams, their contents and technical characteristics,

c) A web server module (implemented as an IIS extension) that handled requests for manifests and fragments, and in the case of live streaming handled real-time ingest of fragments and dynamic generation of client manifests, and

d) A client module, developed initially for Microsoft Silverlight applications but later also made available for other app development frameworks, that dynamically adapted playback quality according to real-time network conditions and CPU performance.

64. Smooth Streaming media streams were described in two complementary manifest files:

a) A server manifest file (.ism), created during encoding, which defined the available streams, codecs, bitrates, and file associations.

b) A client manifest file (.ismc), generated either at encoding time (for VOD) or dynamically by the Smooth Streaming server at playback time (for live). This XML document described the structure of the presentation, including stream information, stream types (video, audio, captions), available quality levels, and fragment durations.

65. Smooth Streaming implemented adaptive streaming by dividing encoded media into small, independent segments – typically 2-5 seconds in duration – each stored as a movie fragment within a single contiguous MP4 file. The client requested these fragments sequentially, adjusting its selection of bitrate levels in response to real-time measurements of bandwidth, CPU utilization, and buffer status.

66. After interpreting the client manifest, clients would make requests for fragments by issuing HTTP requests using RESTful URLs that referenced a fragment's quality level and timestamp. For example:

`http://server/video.ism/QualityLevels(1000000)/Fragments(video=180000000)`

In this example URL, 1000000 represented the encoded bitrate in bits per second, and 180000000 indicated the starting timestamp of the fragment (in 100-nanosecond units).

67. The IIS Smooth Streaming server extension – implemented as a native IIS module – responded to the HTTP clients requests by referencing the server manifest to locate the MP4 file associated with a specific media quality level, locating the specific fragment within the MP4 file by looking up its byte range in an index of timestamps, and finally extracting the appropriate byte range from the MP4 file and returning it to the client it in the body of the HTTP response.

68. This stateless, request-response mechanism allowed Smooth Streaming to function transparently through existing HTTP caches and CDNs, vastly improving scalability compared to earlier streaming architectures.

D. Media File Structure

69. Smooth Streaming media was stored in fragmented MP4 files conforming to the ISO Base Media File Format standard (ISO/IEC 14496-12). Microsoft's extensions to ISOBMFF, which included support for content encryption and digital rights management (DRM), were formally specified as Protected Interoperable File Format (PIFF).

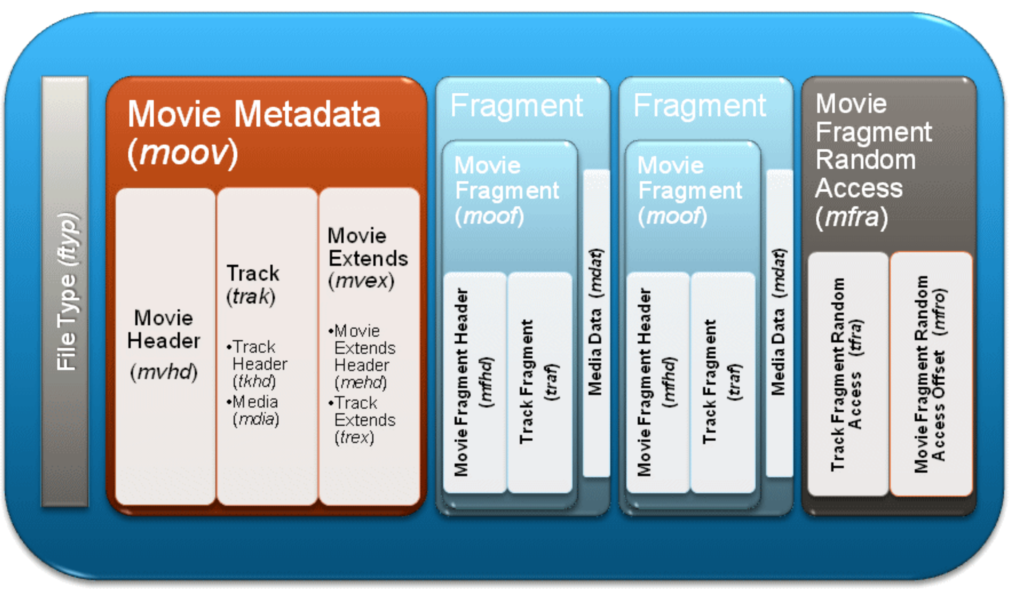

Figure 4: Smooth Streaming file format (PIFF) structure [9][15]

A typical Smooth Streaming media file contained:

a) Movie and track metadata boxes describing media tracks contained in the file,

b) A sequence of *moof* (movie fragment) boxes, each followed by a corresponding *mdat* (media data) box containing the encoded media samples for a short time interval,

c) An optional *mfra* (movie fragment random access) box to facilitate accurate seeking and fast start operations.

70. Each movie fragment contained complete decoding context information so that it could be requested and decoded independently of other fragments. This made it possible to deliver adaptive segments without re-encoding or re-multiplexing at the server.

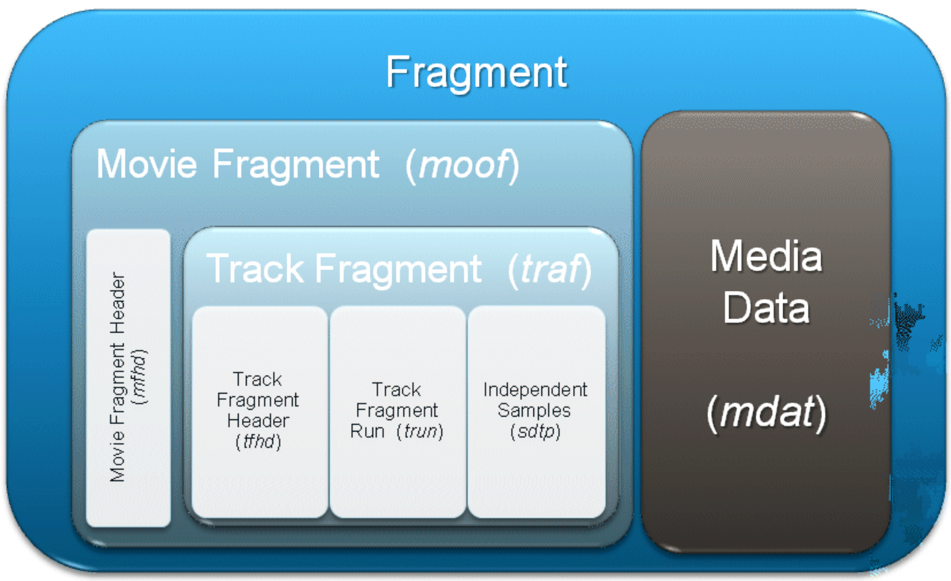

Figure 5: Smooth Streaming fragment (wire format) structure [9][15]

E. Content Encoding and Packaging

71. Smooth Streaming content preparation began with multi-bitrate encoding, producing synchronized streams at several quality levels. During the 2008–2009 timeframe, Expression Encoder was Microsoft's primary product for encoding Smooth Streaming

content, though many encoding software vendors such as Inlet Technologies, Digital Rapids, Telestream, Rhozet and Envivio also implemented support for the format.

72. Expression Encoder performed the following key tasks:

a) Transcoded the source video into multiple bitrates using VC-1 or H.264 codecs, ensuring identical GOP lengths and frame alignment across all renditions,

b) Transcode the source audio into one or more bitrates using WMA or AAC codecs, ensuring sample alignment across all renditions,

c) Packaged each bitrate into a fragmented MP4 file, creating the necessary *moof*, *mdat* and other box structures, and

d) Generated a corresponding *.ism* server manifest file describing all available streams.

73. The alignment of Group-of-Pictures (GOP) boundaries across all video bitrates was mandatory. This ensured that the client could switch bitrates only at keyframe-aligned points, guaranteeing seamless transitions without visual artifacts or decoding errors.

74. The filename extensions used for Smooth Streaming file assets were:

a) *.ismv* – files containing only video or combined video+audio (MP4/PIFF)

b) *.isma* – files containing only audio (MP4/PIFF)

c) *.ism* – server manifest file (XML)

d) *.ismc* – client manifest file (XML)

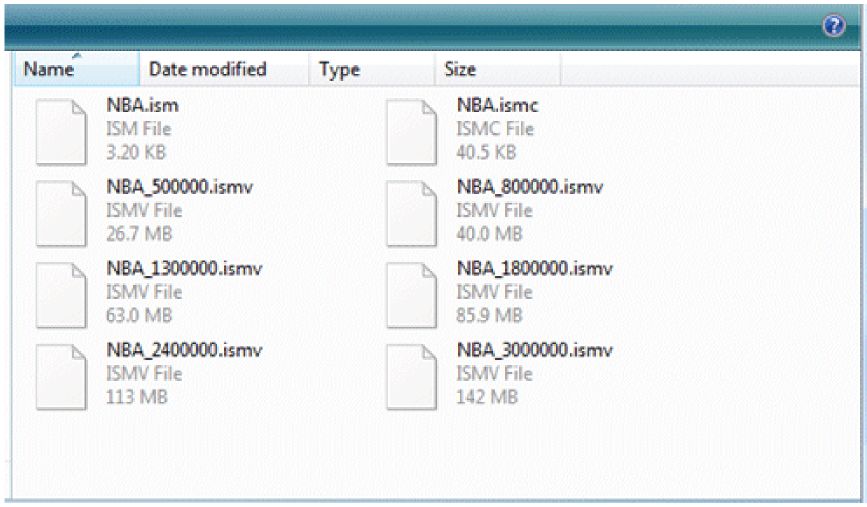

Figure 6: Files of a Smooth Streaming media presentation [9][15]

75. Expression Encoder also supported automatic creation of a simple a Silverlight-based web player capable of parsing Smooth Streaming manifests and demonstrating adaptive streaming playback.

F. Server Architecture

76. The IIS Smooth Streaming server module was implemented as an extension to IIS 7.0 using the IIS integrated request pipeline. It operated at the HTTP layer, interpreting client requests for manifests and fragments and resolving them to specific byte offsets within the underlying MP4 file.

77. When a client requested the Smooth Streaming presentation URL (e.g. `http://server/video.ism/manifest`), the module parsed the .ism server manifest and served the corresponding client manifest (static or dynamically generated). Subsequent requests for fragments used the RESTful syntax described earlier.

78. The module used range-based I/O operations to extract the requested movie fragment directly from disk without copying or re-encoding the media. Because each fragment was stateless and self-contained, the server did not maintain per-user session data or stream

buffers. This architecture made Smooth Streaming horizontally scalable, allowing thousands of concurrent client requests to be handled efficiently by standard IIS caching mechanisms and load-balancing configurations.

79. Smooth Streaming architecture was compatible with HTTP proxies and content delivery networks such as Akamai, Level 3, and Limelight Networks, which cached fragments based on URL uniqueness. The deterministic URL format meant that edge caches could store and serve fragments without repeatedly fetching them from the origin server.

G. Client Architecture

80. The Smooth Streaming client for Silverlight applications was implemented as a Silverlight component known as the Smooth Streaming Media Element (SSME). This component extended Silverlight's built-in MediaElement class, adding functionality to interpret Smooth Streaming manifests, request media fragments over HTTP, and manage adaptive switching during playback.

81. Key responsibilities of the client included:

a. Downloading the client manifest to determine available streams, quality levels (bitrates) and fragment durations;

b. Measuring current download throughput and CPU performance;

c. Estimating optimal bitrate selection and issuing HTTP requests for upcoming fragments;

d. Maintaining target playback buffer levels to prevent underflow;

e. Handling seamless transitions between bitrates at key frame boundaries.

82. At start-up, the client would issue an HTTP request to retrieve the client manifest file. This manifest described characteristics of the media content presentation – including available quality levels (bitrates), fragment (chunk) durations, stream types (video, audio,

text), timescale units, and for live streaming scenarios attributes such as lookahead count (live fragment availability delay introduced by the server) and DVR window length (maximum possible duration of a live client manifest). Having parsed the manifest, the client would select an initial bitrate and issue HTTP requests for fragments. These fragments corresponded to discrete time intervals (typically 2-5 seconds in length), and contained independently decodable media samples. [30][31]

83. The client would begin playback quickly by using a conservative bitrate that it determined was likely to be sustainable based on either a new estimate or data from past playback sessions. As download performance measurements became available during early playback, the client could then refine its choice and either upgrade or downgrade the bitrate accordingly. [31][32]

84. The adaptive-bitrate switching logic resided entirely on the client side. As playback commenced, the Smooth Streaming client monitored recent download throughput, buffer levels, and decoding performance (including CPU utilization). Based on this monitoring, the client could choose to increase or decrease the target bitrate for the next fragment request in order to maximize quality while avoiding buffer underruns and decoding stalls. When it decided to make a switch, the client simply issued a fragment request for the next time interval at the new bitrate level defined in the manifest URIs. Because keyframe and fragment boundary alignment across bitrates was enforced at encoding time, the transition would be seamless and did not require server-side state or session renegotiation. [31]

85. The Smooth Streaming client strived to build a playback buffer by storing multiple downloaded fragments in its internal memory before playing them back, and its algorithm

aimed to maintain these buffer levels (measured as total media duration of fragments buffered) within a safe operating range and used deviations from that range to trigger bitrate adjustments. The main purpose of the buffer was to prevent or minimize disruptions to playback that could be caused if the player ran out of fragments to play because the rate of fragment download couldn't keep up with the rate of playback (which was typically equivalent to clock rate, yielding a 1.0x playback speed). [15][31][32]

a. **Buffer Depletion**: When measured download throughput decreased or fragment downloads began to lag, the buffer began to deplete. The client could monitor the rate of buffer drain by observing changes in buffer level over time, and could determine if the total media duration of fragments buffered was diminishing at a rate higher than the rate at which fragments needed to be played.

   If the rate of buffer drain exceeded a critical high threshold (i.e. buffer was depleting too quickly) or the buffer level approached a critical low threshold (i.e. not enough fragments remained in the buffer to guarantee glitch-free playback), the algorithm lowered the active bitrate selection to a lower quality level (lower bitrate). The goal of the bitrate downgrade was to reduce future fragment sizes and download times, which ideally would help slow down or even reverse the rate of buffer depletion.

b. **Buffer Expansion**: Conversely, if the buffer level remained consistently near its upper threshold and download throughput measurements exceeded the current bitrate by a significant margin, the client inferred that additional bandwidth was available. After a stabilization period the client could then upgrade the active bitrate selection to a higher quality level.

86. Seeking within a presentation was handled via fragment indexing and URL translation. When the user jumped to a new playback time (either forward or backward), the client computed the fragment index corresponding to the target playback time based on the fragment duration and timescale units declared in the manifest. The client then requested the fragment at the new timestamp (and appropriate bitrate level) by constructing a RESTful URL. The server would map the request URL to the underlying media file and return the correct fragment bytes via HTTP. Because the manifest and underlying files were encoded with aligned GOPs and indexed fragment boundaries, the client's seek point would begin decoding cleanly without depending on decoding any earlier fragments. [9][15]

87. These behaviors were consistent with the client-side adaptive bitrate logic described in patents US 7,925,774 (Zhang), US 7,949,775 (Virdi) and US 8,370,887 (Virdi) which specified buffer-based thresholds as key decision variables in client-side bitrate adaptation.

H. Live Smooth Streaming and DRM Support

88. Although Smooth Streaming was first designed for VOD delivery, the architecture was extended in early 2009 to support live streaming. In this mode, an encoder continuously generated new MP4 fragments and pushed them to the IIS server using a publish point configuration. The server automatically updated the manifest as new fragments arrived, and clients periodically refreshed their manifests to detect new segments. The live ingestion pipeline supported multiple simultaneous encoders, each representing a different bitrate ladder. The IIS server synchronized fragment timestamps across bitrates to preserve alignment and enable adaptive switching in real time.

89. While early 2009 Smooth Streaming content was primarily unencrypted, Microsoft subsequently integrated Smooth Streaming with its PlayReady DRM technology. Because the Smooth Streaming format was based on Protected Interoperable File Format (PIFF), encryption and license signaling could be embedded using standard MP4 boxes without altering the delivery mechanism.

I. Other Product Integrations

90. Smooth Streaming also integrated with other Microsoft media technologies, including:

a. Silverlight Media Framework, for building customized players;

b. IIS Transform Manager, for server-side media processing automation;

c. Windows Azure Media Services (introduced later), which adopted Smooth Streaming as its default delivery format and eventually replaced IIS Media Services as Microsoft's flagship media server product.

J. Impact and Legacy

91. IIS Smooth Streaming represented a fundamental redesign of Microsoft's streaming architecture. It replaced proprietary streaming protocols with an open, HTTP-based approach built on standard MP4 fragmentation and client-side intelligence. The resulting system combined the scalability of web delivery with the quality of traditional broadcast experiences.

92. By formalizing concepts first prototyped and validated during the 2008 Beijing Olympics, Microsoft established an end-to-end platform that bridged encoding, packaging, delivery, and playback. The Smooth Streaming model – manifest-driven, HTTP-delivered, and client-adaptive – formed the conceptual and technical foundation for later streaming standards such as MPEG DASH and for the widespread adoption of adaptive bitrate delivery across the industry.

I, Alex Zambelli, declare under penalty of perjury that the foregoing is true and correct.

Dated: October 24, 2025

Alex Zambelli

Appendix: MATERIALS CONSIDERED

[1] G. Johnston, "ABC.com Unveils 'HD' Video Player," *Broadcasting & Cable*, Apr. 2, 2007. [Online]. Available: https://web.archive.org/web/20070509114928/http://www.broadcastingcable.com/article/CA6429850.html. [Accessed: Oct. 19, 2025].

[2] BeetTV, "ABC.com Readies Online HD Shows as 'Streamlettes' from Move," *YouTube*, May 22, 2007. [Online video]. Available: https://www.youtube.com/watch?v=sYk9eVS8Sck. [Accessed: Oct. 19, 2025].

[3] National Association of Broadcasters, "High Definition Internet Streaming Video," *TV TechCheck*, Jul. 30, 2007. [Online]. Available: https://www.nab.org/xert/scitech/pdfs/tv073007.pdf. [Accessed: Oct. 19, 2025].

[4] BeetTV, "John Edwards, Move Networks," *YouTube*, Jul. 30, 2008. [Online video]. Available: https://www.youtube.com/watch?v=2yl5XLHr_w0. [Accessed: Oct. 19, 2025].

[5] A. Zambelli, "An Inside Look at NBC Olympics Video Player," *Alex Zambelli's Streaming Media Blog*, Aug. 21, 2008. [Online]. Available: https://alexzambelli.com/blog/2008/08/21/an-inside-look-at-nbc-olympics-video-player/. [Accessed: Oct. 19, 2025].

[6] A. Zambelli, "A Brief History of Multi-Bitrate Streaming," *Alex Zambelli's Streaming Media Blog*, Dec. 17, 2008. [Online]. Available: https://alexzambelli.com/blog/2008/12/17/a-brief-history-of-multi-bitrate-streaming/. [Accessed: Oct. 19, 2025].

[7] J. Bocharov, "I've Seen the Future and the Future is… Smooth!" *IIS.net Blogs*, Oct. 28, 2008. [Online]. Available:

https://web.archive.org/web/20090605235019/http://blogs.iis.net/jboch/archive/2008/10/28/i-ve-seen-the-future-and-the-future-is-smooth.aspx. [Accessed: Oct. 19, 2025].

[8] A. Zambelli, "The Birth of Smooth Streaming," *Alex Zambelli's Streaming Media Blog*, Feb. 4, 2009. [Online]. Available: https://alexzambelli.com/blog/2009/02/04/the-birth-of-smooth-streaming/. [Accessed: Oct. 19, 2025].

[9] A. Zambelli, "Smooth Streaming Architecture," *Alex Zambelli's Streaming Media Blog*, Feb. 10, 2009. [Online]. Available: https://alexzambelli.com/blog/2009/02/10/smooth-streaming-architecture/. [Accessed: Oct. 19, 2025].

[10] J. Bocharov, "Beta Release of Smooth Streaming Now Available," *IIS.net Blogs*, Feb. 24, 2009. [Online]. Available: https://web.archive.org/web/20090604163237/http://blogs.iis.net/jboch/archive/2009/02/24/beta-release-of-smooth-streaming-now-available.aspx. [Accessed: Oct. 19, 2025].

[11] A. Zambelli, "Smooth Streaming Beta Released," *Alex Zambelli's Streaming Media Blog*, Feb. 25, 2009. [Online]. Available: https://alexzambelli.com/blog/2009/02/25/smooth-streaming-beta-released/. [Accessed: Oct. 19, 2025].

[12] J. Bocharov, "Delivering Media with IIS Media Services," presented at MIX09 Conference, Las Vegas, NV, USA, Mar. 18–20, 2009. [Online]. Available: https://web.archive.org/web/20090426001804/http://mschannel9.vo.msecnd.net/o9/mix/09/pptx/t56f.pptx. [Accessed: Oct. 19, 2025].

[13] A. Zambelli, "Microsoft Silverlight Media End-to-End," presented at MIX09 Conference, Las Vegas, NV, USA, Mar. 18–20, 2009. [Online]. Available: http://web.archive.org/web/20090411041129/http://mschannel9.vo.msecnd.net/o9/mix/09/pptx/t43f.pptx. [Accessed: Oct. 19, 2025].

[14] A. Zambelli, "Microsoft Silverlight Media End-to-End," presented at MIX09 Conference, Las Vegas, NV, USA, Mar. 18–20, 2009. [Online video]. Available: http://web.archive.org/web/20090411041129/http://mschannel9.vo.msecnd.net/o9/mix/09/wmv-hq/t43f.wmv. [Accessed: Oct. 19, 2025].

[15] A. Zambelli, "IIS Smooth Streaming Technical Overview," Microsoft, Redmond, WA, USA, Mar. 25, 2009. [Online]. Available: https://web.archive.org/web/20090402042137/http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=03d22583-3ed6-44da-8464-b1b4b5ca7520. [Accessed: Oct. 19, 2025].

[16] B. Waggoner, "NAB Day 1: Smooth Streaming Released, 1080p in Silverlight, New VC-1 and More," *On10.net Blogs*, Apr. 20, 2009. [Online]. Available: https://web.archive.org/web/20090501182432/http://on10.net/blogs/benwagg/NAB-Day-1-Smooth-Streaming-released-1080p-in-Silverlight-new-VC-1-and-more/. [Accessed: Oct. 19, 2025].

[17] Microsoft Corporation, "Microsoft Smooth Streaming Provides True High-Definition Video Delivery," *Microsoft News Center*, Apr. 20, 2009. [Online]. Available: https://web.archive.org/web/20090501182432/http://www.microsoft.com/presspass/press/2009/apr09/04-20SmoothStreamingPR.mspx. [Accessed: Oct. 19, 2025].

[18] Silverlight Team, "French International Tennis Tournament at Roland Garros Broadcast Live on the Web Using IIS 7 Smooth Streaming," *Silverlight.net Blog*, Jun. 9, 2009. [Online]. Available: https://web.archive.org/web/20091107070314/http://team.silverlight.net/lighthouse-wins/french-

international-tennis-tournament-at-roland-garros-broadcast-live-on-the-web-using-iis-7-smooth-streaming/. [Accessed: Oct. 19, 2025].

[19] Silverlight Team, "Wimbledon is Here – We're Giving the Championships to You Live," *Silverlight.net Blog*, Jun. 25, 2009. [Online]. Available: https://web.archive.org/web/20091105141851/http://team.silverlight.net/lighthouse-wins/wimbledon-is-here-and-we-rsquo-re-giving-the-championships-to-you-live/. [Accessed: Oct. 19, 2025].

[20] Silverlight Team, "Michael Jackson: The King of Pop – A Memorial Live in HD," *Silverlight.net Blog*, Jul. 7, 2009. [Online]. Available: https://web.archive.org/web/20091030163134/http://team.silverlight.net/announcement/michael-jackson-the-king-of-pop-a-memorial-ndash-live-in-hd/. [Accessed: Oct. 19, 2025].

[21] J. Ozer, "Streaming Gets Smarter: Evaluating the Adaptive Streaming Technologies," *Streaming Media*, Jul. 31, 2009. [Online]. Available: https://www.streamingmedia.com/Articles/Editorial/Featured-Articles/Streaming-Gets-Smarter-Evaluating-the-Adaptive-Streaming-Technologies-65544.aspx. [Accessed: Oct. 19, 2025].

[22] Silverlight Team, "Sunday Night Football Powered by Silverlight," *Silverlight.net Blog*, Aug. 20, 2009. [Online]. Available: https://web.archive.org/web/20091030163249/http://team.silverlight.net/lighthouse-wins/sunday-night-football-powered-by-silverlight/. [Accessed: Oct. 19, 2025].

[23] A. Zambelli, "NBC Sunday Night Football Powered by Silverlight and Smooth Streaming," *Alex Zambelli's Streaming Media Blog*, Sep. 10, 2009. [Online]. Available: https://alexzambelli.com/blog/2009/09/10/nbc-sunday-night-football-powered-by-silverlight-and-smooth-streaming/. [Accessed: Oct. 19, 2025].

[24] A. Zambelli, "IIS Media Services 3.0 and Player SDK Released," *Alex Zambelli's Streaming Media Blog*, Oct. 12, 2009. [Online]. Available: https://alexzambelli.com/blog/2009/10/12/iis-media-services-3-0-and-player-sdk-released/. [Accessed: Oct. 19, 2025].

[25] A. Wallenstein, "Microsoft and NBC Universal partner on Olympics," *Reuters*, Jan. 7, 2008. [Online]. Available: https://www.reuters.com/article/lifestyle/microsoft-and-nbc-universal-partner-on-olympics-idUSN07371119/. [Accessed: Oct. 19, 2025].

[26] A. Kinney, E. Schmidt, "Building NBCOlympics.com with Silverlight," *Channel 9*, Aug. 7, 2008. [Online video]. Available: https://web.archive.org/web/20080908055952/http://mschnlnine.vo.llnwd.net/d1/ch9/9/2/9/9/1/4/TCS2NBCOlympics_2MB_ch9.wmv. [Accessed: Oct. 19, 2025].

[27] E. Lai, "NBC scored Olympic gold with Silverlight, but online ads failed to place," *Computer World*, Sep. 9, 2008. [Online]. Available: https://www.computerworld.com/article/1580543/nbc-scored-olympic-gold-with-silverlight-but-online-ads-failed-to-place.html. [Accessed: Oct. 19, 2025].

[28] Microsoft Corporation, "Microsoft and Akamai Innovate on Consumer Video Experiences Using Silverlight," *Yahoo! Finance*, Oct. 28, 2008. [Online]. Available: https://web.archive.org/web/20081105112604/http://biz.yahoo.com/prnews/081028/aqtu537.html?.v=26. [Accessed: Oct. 19, 2025].

[29] T. Siglin, "Akamai and Microsoft Attempt to Stream HD Smoothly," *Streaming Media*, Oct. 28, 2008. [Online]. Available: https://www.streamingmedia.com/Articles/Editorial/Featured-Articles/Akamai-and-Microsoft-Attempt-to-Stream-HD-Smoothly-65328.aspx. [Accessed: Oct. 19, 2025].

[30] Microsoft Corporation, "IIS Smooth Streaming Client Manifest - SmoothStreamingMedia Element," *Microsoft Learn,* May 1, 2012. [Online]. Available: https://learn.microsoft.com/en-us/previous-versions/iis/smooth-streaming-client/iis-smooth-streaming-client-manifest-smoothstreamingmedia-element. [Accessed: Oct. 20, 2025].

[31] J. Ghosh, "IIS Smooth Streaming - Enhancing Silverlight Video Experiences with Contextual Data," *Microsoft Learn – MSDN Magazine*, Mar. 2010. [Online]. Available: https://learn.microsoft.com/en-us/archive/msdn-magazine/2010/march/iis-smooth-streaming-enhancing-silverlight-video-experiences-with-contextual-data. [Accessed: Oct. 20, 2025].

[32] J. Jiang, V. Sekar, H. Zhang, "Improving Fairness, Efficiency, and Stability in HTTP-based Adaptive Video Streaming with FESTIVE," *CoNEXT'12*, Dec. 10-13, 2012. [Online]. Available: https://www.andrew.cmu.edu/user/vsekar/papers/conext12_festive.pdf. [Accessed: Oct. 20, 2025].

[33] A. Zambelli, "Silverlight 3, IIS Media Services 3.0, Olympics 2010 – Wow, It Truly Is March Madness!" *Alex Zambelli's Streaming Media Blog*, Mar. 19, 2009. [Online]. Available: https://alexzambelli.com/blog/2009/03/19/silverlight-3-iis-media-services-30-olympics-2010-wow-it-truly-is-march-madness/. [Accessed: Oct. 20, 2025].

[34] *IIS Smooth Streaming Transport Protocol*, [MS-SMTH] - v20090908, Microsoft Corporation, Redmond, WA, USA, Sep. 8, 2009. [Online]. Available: https://web.archive.org/web/20120907004742/https://www.iis.net/community/files/media/smoothspecs/[MS-SMTH].pdf. [Accessed: Oct. 20, 2025].

[35] BeetTV, "News From NAB: Microsoft Rolls Out "Adaptive Streaming" for HD Video," *YouTube*, Apr. 23, 2009. [Online video]. Available: https://www.youtube.com/watch?v=5U0TPv9Bc1k. [Accessed: Oct. 22, 2025].

[36] C. Knowlton, “Live Broadcasting with Silverlight and Windows Media,” presented at Streaming Media West conference, San Jose, CA, USA, Nov. 17-19, 2009. [Online]. Available: https://download.microsoft.com/download/a/a/b/aabeb56e-45b7-4145-bdb1-595de745f61f/smwest09-msft-workshop.pdf [Accessed: Oct. 23, 2025].

[37] Microsoft Corporation, “Silverlight 2 Technical Reviewers Guide,” Microsoft, Redmond, WA, USA, Oct. 2008. [Online]. Available: https://download.microsoft.com/download/8/e/4/8e4a41a0-4b29-49b6-8e17-0da540f27d1b/Silverlight_2_Reviewers_Guide.pdf. [Accessed: Oct. 23, 2025].

[38] Microsoft Corporation, “Delivering Live and On-Demand Smooth Streaming,” Microsoft, Redmond, WA, USA, 2009. [Online]. Available: https://download.microsoft.com/download/4/E/5/4E599FBB-6E34-4A74-B3C5-1391CB0FD55F/Delivering_Live_and_On-Demand_Smooth_Streaming.pdf. [Accessed: Oct. 23, 2025].

[39] Microsoft Corporation, “Live and On-Demand Video with Silverlight and IIS Smooth Streaming,” Microsoft, Redmond, WA, USA, Feb. 2010. [Online]. Available: https://download.microsoft.com/download/3/A/4/3A4A066C-6543-4BC1-A8BA-965D0E55F296/Live_and_On_Demand_Video_with_Silverlight_and%20IIS_Smooth_Streaming_FINAL.pdf. [Accessed: Oct. 23, 2025].

Appendix: Resume

az@alexzambelli.com
+1 (425) 444-5363
www.linkedin.com/in/alexzambelli

ALEX ZAMBELLI

SUMMARY

Streaming media industry professional with over 23 years of experience in product management and engineering, deep technical knowledge of digital media technologies, cross-disciplinary communication skills, agile software development proficiency, and a passion for video & audio.

SKILLS

- Video on-demand (VOD), live linear and live event streaming workflows
- Video encoding & packaging for adaptive bitrate (ABR) streaming
- ABR streaming protocols and formats (MPEG DASH, HLS, CMAF)
- Video and audio compression (H.264, H.265, AAC, AC-4, DD+)
- High dynamic range video (Dolby Vision, HDR10, HLG)
- Playback quality-of-experience (QoE) instrumentation and analytics
- Open-source media processing tools such as FFmpeg, GPAC, AviSynth

EXPERIENCE

SENIOR PLATFORM MANAGER

DOLBY LABORATORIES

November 2023 – present Seattle, WA (remote)

Core technology platform manager in Dolby's Entertainment organization.

Responsible for managing Dolby Vision core components used in video encoding workflows, and for fostering adoption of Dolby video and audio formats in OTT content distribution by managing development of Dolby OTT specifications, test content, content tools and playback test apps.

TECHNICAL PRODUCT MANAGER IV

WARNER BROS. DISCOVERY

September 2021 – October 2023 Bellevue, WA

Technical product manager in DTC Global Video Platform product group, responsible for product managing media processing platforms that support Max and Discovery+ streaming services.

- Defined VOD mezzanine, transcoding and packaging requirements for Max, WBD's new flagship streaming service launched in 2023
- Drove improvements in Max video quality (HFR, 10-bit SDR, quality-based encoding)
- Designed Discovery+ solution for delivering compatible DASH/HLS content manifests based on playback device capabilities

PRINCIPAL PRODUCT MANAGER

HULU

August 2016 – August 2021 Seattle, WA

Product manager for VOD and live video platform - from ingest to transcoding to publishing to playback. Responsible for managing strategy, design and deployment of new audio/video features to Hulu content and devices.

- Drove deployment of 5.1 surround sound, audio descriptions, loudness normalization, HFR, 4K UHD and HDR video to Hulu VOD platform
- Initiated and achieved platform-wide adoption of HEVC video codec for Hulu VOD
- Proposed and oversaw development of Hulu's universal HDR streaming format
- Defined requirements for new Live TV player UX (2017), including features specific to live linear playback, startover rights, DVR playback, ad controls, program extensions, etc.
- Wrote specifications for transport and encoding of live linear video by signal acquisition vendors and broadcast network partners
- Led effort to standardize, unify and centralize handling, reporting and viewer messaging of playback errors in Hulu apps
- Defined QoS/QoE KPIs, goals and optimization strategies for VOD and live playback

PRINCIPAL PRODUCT MANAGER

ISTREAMPLANET (subsidiary of **TURNER BROADCASTING**)

January 2013 – August 2016 Redmond, WA

Product manager for Aventus, iStreamPlanet's cloud-based live video encoding platform, used by broadcasters such as NBC, Turner and Fox to stream live linear channels as well as major international events such as the Olympics (2014, 2016) and NFL Super Bowl (2015). The success of Aventus contributed to iStreamPlanet's acquisition by Turner Broadcasting in 2015.

- Defined Aventus core media processing requirements; designed UX components and APIs; wrote technical documentation and product release notes
- Built video encoding profiles for ABR streaming; optimized codec settings
- Served as scrum product owner, technical liaison to partners and customers, and principal video specialist for the company

SENIOR TECHNICAL EVANGELIST

MICROSOFT

July 2007 – December 2012 Redmond, WA

- Evangelized Microsoft Media Platform technologies (including Azure/IIS Media Services, Smooth Streaming, Silverlight, PlayReady, VC-1) through tech community outreach and consulting on high-profile customer projects
- Served as solutions architect on major live streaming projects such as 2012 London Olympics (CTV/FTV/RTVE), 2010 Vancouver Olympics (NBC/CTV), 2008 Beijing Olympics (NBC), Sunday Night Football (NBC) and NASCAR RaceBuddy 3D (TNT)
- Championed HTTP-based adaptive streaming technology, among first in the industry

SOFTWARE DEVELOPMENT ENGINEER IN TEST

MICROSOFT

September 2002 – June 2007 Redmond, WA

- Responsible for testing VC-1 (WMV9) DXVA decoder for Windows Vista, and VC-1 encoder for various partner releases
- Tested Windows Media Player v9 and v10 releases for Windows desktop and mobile

---

EDUCATION **FLORIDA INSTITUTE OF TECHNOLOGY** – MELBOURNE, FL

Bachelor of Science in **Computer Science** with Honors, 2002

SELECTED PUBLICATIONS

A Brief History of Multi-Bitrate Streaming. (2008). *AlexZambelli.com*. https://alexzambelli.com/blog/2008/12/17/a-brief-history-of-multi-bitrate-streaming/

IIS Smooth Streaming Technical Overview. (2009). *Microsoft*. https://alexzambelli.com/archive/IIS_Smooth_Streaming_Technical_Overview.pdf

A History of Media Streaming and the Future of Connected TV. (2013). *The Guardian*. https://www.theguardian.com/media-network/media-network-blog/2013/mar/01/history-streaming-future-connected-tv

Cord-cutting: Beginning of the End for Linear Television. (2014). *The Guardian*. https://www.theguardian.com/media-network/media-network-blog/2014/jul/07/cord-cutting-internet-tv-netflix

It's Time We Said Goodbye to Fractional Framerates. (2022). *Demuxed*. https://www.youtube.com/watch?v=RDHgSbmJvjs

---

LITIGATION CONSULTING EXPERIENCE

Litigation consultant and fact witness for defense counsel in Adeia Guides Inc. v. BCE Inc. (Canada Federal Court File No. T-1184-21)

EXHIBIT 2
REDACTED IN ITS ENTIRETY

EXHIBIT 3

Trials@uspto.gov
571-272-7822

Paper 16
Entered: March 4, 2024

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

PELOTON INTERACTIVE, INC.,
Petitioner,

v.

NEC CORPORATION,
Patent Owner.

IPR2023-01239
Patent 8,752,101 B2

Before MICHAEL J. STRAUSS, CHARLES J. BOUDREAU, and CHRISTOPHER L. OGDEN, *Administrative Patent Judges*.

STRAUSS, *Administrative Patent Judge*.

DECISION
Granting Institution of *Inter Partes* Review
35 U.S.C. § 314

(iv) Available Reproduction Time

Patent Owner contends Wang's variable "T_{last}" fails to teach an available reproduction time because, in those cases in which packets arrive out of order, the "time available for reproducing the content based on the content data stored in the storage device" will be different than the "time difference between the current time and the scheduled playout time of the last packet in the buffer." Prelim. Resp. 36–37 (citing Ex. 2001 ¶¶ 81, 82, 88 (internal citations omitted)). Thus, Patent Owner argues "Wang does not disclose 'an available reproduction time' as claimed." *Id.* at 37 (citing Ex. 2001 ¶ 82). Patent Owner further contends Wang's T_{last} is equal to the playout time of the last packet from the current time and includes the time before the buffer playout begins and, therefore, is not the available reproduction time but can be something greater. *Id.* at 42 (citing Ex. 2001 ¶ 88 (internal citations omitted)).

c) Analysis

Having considered the parties' respective arguments and the cited evidence, we agree with Patent Owner that Petitioner has not shown a reasonable likelihood that it would prevail on its allegations regarding claim 1 under Ground 2.

(i) Code Rate

First, at this preliminary stage, we are not persuaded by Petitioner's contentions that Wang teaches [1d]'s code rate. *See* Pet. 62–65; *see also* Prelim. Resp. 28. The '101 patent discloses "[t]he code rate represents the data amount (the data size, e.g., the number of bytes) of content data for reproducing content at constant speed (reproducing content at 1x speed) for a unit time (e.g., one second)." Ex. 1001, 7:7–11. In contrast, we agree with

Patent Owner that Wang's "'transmission rate' is the number of packets that the sender sends per unit of time" and "refers to the rate of packet transmission, not to changes to the code allowing for distinct code rates as required by the Challenged Claims of the '101 Patent." Prelim. Resp. 28; *see also id.* at 39–41.

(ii) Motivations to Combine

Second, at this preliminary stage, we are not persuaded by Petitioner's contentions that a POSITA would have been motivated to combine the teachings of Chou and Wang. Pet. 74. As argued by Patent Owner, "Wang and Chou are directed to different technologies—active encoding applications and transport-layer protocols," (*see* Prelim. Resp. 30, 34–35), and Petitioner does not persuasively explain, for example, why the two methods would be combined into a single method rather than implemented independently in their respective transport and application layers.

(iii) Scope of Wang's Teaching vis-à-vis Limitation [1d]

Petitioner states "[i]f [Patent Owner] argues Chou doesn't teach [1d], Wang does" (Pet. 62), concluding "Wang teaches calculating a change in code rate based on both the remaining time before reproduction start time and an available reproduction time, as required by [1d]" (*id.* at 63). Apart from a bare cross-reference to Ground 1 (*see* Pet. 62 ("*See* Ground 1, [1d]")), Petitioner does not make any reference to Chou for teaching any aspect of limitation [1d] in connection the combination of Ground 2 (*see generally id.* at 62–65). Furthermore, although contending that "Chou and Wang both disclose controlling *code rate* to control buffer duration/fullness to prevent buffer underflow" (*id.* at 72), Petitioner does not clearly indicate

that a POSITA would have had reason to combine Wang's teaching of criteria used to maintain buffer fullness with Chou's teaching of determining a code rate where the code rate "represents the data amount (the data size, e.g., the number of bytes) of content data for reproducing content at constant speed (reproducing content at 1x speed) for a unit time (e.g., one second)" (Ex. 1001, 7:7–11). Therefore, at this preliminary stage, we treat Petitioner's Ground 2 as relying on Wang alone for teaching [1d].

(iv) Available Reproduction Time

Patent Owner contends that Wang's variable "T_{last}" fails to teach an available reproduction time is based on those cases in which packets arrive out of order. *See* Prelim. Resp. 36–37. However, Patent Owner does not explain why, in other cases wherein, for example, the packets do arrive in order, "T_{last}" does teach an available reproduction time or why a POSITA would not have made appropriate adjustments to account for such cases. In an obviousness analysis, it is not necessary to find precise disclosure directed to the specific subject matter claimed because inferences and creative steps that a person of ordinary skill in the art would employ can be taken into account. *See KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 418 (2007). In this regard, "[a] person of ordinary skill is also a person of ordinary creativity, not an automaton." *Id.* at 421. Thus, although at this preliminary state of the proceeding, we do not find Patent Owner's contention to be supported by the evidence, the parties will have the opportunity to develop the record on this issue further during trial. Furthermore and nonetheless, because Petitioner's other allegations are insufficient for the reasons set forth *supra*, our ultimate conclusion regarding claim 1 is not affected.

EXHIBIT 4

Trials@uspto.gov
571-272-7822

Paper 43
Entered: March 3, 2025

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

PELOTON INTERACTIVE, INC.,
Petitioner,

v.

NEC CORPORATION,
Patent Owner.

IPR2023-01239
Patent 8,752,101 B2

Before MICHAEL J. STRAUSS, CHARLES J. BOUDREAU, and CHRISTOPHER L. OGDEN, *Administrative Patent Judges*.

STRAUSS, Administrative Patent Judge.

JUDGMENT
Final Written Decision
Determining No Challenged Claims Unpatentable
35 U.S.C. § 318(a)

b) Patent Owner's Contentions

(i) Code Rate

Patent Owner contends Wang fails to teach limitation [1d]'s determining or changing a code rate. PO Resp. 48. According to Patent Owner, "Wang discloses adjusting 'the transmission rate and/or nominal playout rate . . . to maintain the receive[r] buffer fullness level[.]'" *Id.* (internal citations omitted; alterations in original) (quoting Pet. 62). However, according to Patent Owner, Wang's "'transmission rate' is the number of packets that the sender sends per unit of time" and "refers to the rate of packet transmission, not to changes to the code allowing for distinct code rates as required by the Challenged Claims of the '101 Patent." *Id.* at 48. Patent Owner further argues, because "[t]he 'nominal playout rate' is the rate the receiver reproduced the content per unit of time, . . . [it] is unrelated to the code rate of the data as required by the Challenged Claims." *Id.* Therefore, according to Patent Owner, "[n]either [Wang's] transmission nor playback rates suggest determining or changing ***a code rate*** of the underlying data." *Id.*; *see also id.* at 55–57.

(ii) Motivations to Combine

Patent Owner contends "[t]he Petition's arguments for motivations to combine fail because they are conclusory and change the principle of operation of the underlying references." PO Resp. 48 (emphasis omitted). Patent Owner argues a POSITA would not have been motivated to combine the teachings of Chou and Wang based on several arguments:

1. Chou teaches away from determining the code rate based on the available time. PO Resp. 48.

2. "Petitioner fails to show a reasonable expectation of success, as *Wang* and *Chou* are directed to different technologies, deployed at different network layers and operate independently to preserve efficiency and modularity in network system design." *Id.* at 49 (citing Pet. 68–75, Ex. 2012 ¶ 122, Ex. 2006, 313); *see also id.* at 54.
3. "[T]he Petition fails to explain how a POSITA would have substituted or supplemented the proposed 'linear quadratic formula' disclosed by *Chou* (which specifically proclaims that 'it is ***not*** the actual fullness of the client buffer that is controlled') with the elements from *Wang*" and, if combined, "would not (and could not) accept the elements of *Wang* without substantial changes." *Id.* at 52 (citing Ex. 2012 ¶ 127).
4. Combining the asserted element from Wang into Chou would defeat the purpose of Chou by controlling actual buffer fullness rather than using a linear quadratic function to control code rate. *Id.* (citing Ex. 1004 ¶¶ 19, 70, 128).
5. The "combination would change the principle of operation of *Chou* and not be possible without significant effort and experimentation." *Id.* at 52–53 (citing Ex. 1004 ¶ 129). *See also id.* at 53–54.

(iii) Scope of Wang's Teaching vis-à-vis Limitation [1d]

Patent Owner contends the Petition is unclear about what element of Wang is alleged to be incorporated into Chou. PO. Resp. 49–50 (citing Ex. 2012 ¶ 124). According to Patent Owner,

> it is unclear whether the Petition is arguing that *Wang* could theoretically be combined with *Chou* to remedy *Chou's* failure to disclose Claim Element [1d] in its entirety, or if the Petition is arguing that *Wang* could be combined with *Chou* to remedy only a portion of Claim Element [1d].

Id. (citing Pet. 69–75 (§IX.D.1); Ex. 2001 ¶ 72).

Patent Owner argues, "[a]s such, the Petition fails to articulate what combination it is proposing and is therefore deficient." *Id.* at 50. According to Patent Owner "because the Petition does no more than simply point to various independent portions in *Wang* and *Chou* without explaining what is to be modified, why that modification would have been beneficial or motivated by anything other than hindsight, or how the combination would work, the Petition is deficient." *Id.* (citing *Metalcraft of Mayville, Inc. v. Toro Co.*, 848 F. 3d 1358, 1367 (Fed. Cir. 2017)).

(iv) Available Reproduction Time

Patent Owner contends Wang's variable "T_{last}" fails to teach an available reproduction time because, in those cases in which the current last packet arrives before earlier packets in the bitstream, the "time available for reproducing the content based on the content data stored in the storage device" will be smaller than the "time difference between the current time and the scheduled playout time of the last packet in the buffer." PO Resp. 57 (citing Ex. 2012 ¶ 138). Thus, Patent Owner argues, although "in some circumstances the value of 'T_{last}' *may* correspond to a value of 'a time available for reproducing the content based on the content data stored in the storage device of the reception device,' . . . it does not necessarily correspond to the same value." *Id.* at 58 (citing Ex. 2012 ¶ 139). Patent Owner further argues a POSITA would not know how to determine those

situations that would result in a divergence of the values or how to correct for such errors. *Id*

(v) Determining a Code Rate based on Remaining and Available Times

Patent Owner contends "the Petition is deficient in its support and reasoning." PO Resp. 59 (citing Ex. 2012 ¶ 143). Patent Owner argues, although Petitioner asserts Wang's T_x reaches claim 1's remaining time, T_x only represents the time difference between the current and scheduled playout time. *Id.* (citing Ex. 2012 ¶ 143, Pet. 62–63). Patent Owner further disputes Petitioner's mapping of T_{last} for teaching available reproduction time. *Id.* (citing Ex. 2012 ¶ 143, Pet. 63). Patent Owner also asserts "[t]he Petition . . . presents a different theory [under which Wang teaches available reproduction time] via a hypothetical situation created by Petitioner" that lacks sufficient evidentiary support and "is conjecture made up by Petitioner and its expert" and is "nonsensical." *Id.* at 60 (citing Ex. 2012 ¶ 144; Pet. 64–65). Moreover, Patent Owner argues Petitioner's hypothetical relies on unrealistic assumptions such that T_{first}, applied by Petitioner for teaching remaining time, could not be calculated. *Id.* at 60–61.

Patent Owner further contends, even "accepting the assumptions presented by the hypothetical, the *Wang* reference itself contradicts the conclusions made in the Petition and demonstrates why T_{last} is ***not*** the claimed 'available reproduction time.'" PO Resp. 61 (citing Ex. 2012 ¶ 145). In support, Patent Owner provides an example of timings that would result during playback of a first of five packets stored in a buffer asserted to "demonstrate that 'T_{last}' would produce a value that is not representative of

the time available for reproducing the content as the claims require." *Id.* at 61–62 (citing Ex. 1012 ¶ 145, Pet. 64–65).

c) Petitioner's Reply

Petitioner does not address the Wang reference or any other arguments presented by Patent Owner in connection with Petitioner's obviousness assertions of claims 1, 4–6, 10–16, and 19–26 over the combined teachings of Chou and Wang, Petitioner's Ground 2. Pet. Reply.

d) Analysis

In our Institution Decision, although expressing some uncertainty with respect to certain of Patent Owner's arguments on the preliminary record then before us, "we [were] not persuaded by Petitioner's contentions that [(1)] Wang teaches [1d]'s code rate" (Dec. 53 (citing Pet. 62–65) or (2) "a POSITA would have been motivated to combine the teachings of Chou and Wang" (*id.* at 54 (citing Pet. 74)). Petitioner does not dispute those determinations in its Reply or otherwise. Furthermore, Petitioner does not address or otherwise dispute Patent Owner's arguments challenging the asserted obviousness of claims 1, 4–6, 10–16, and 19–26 over the combined teachings of Chou and Wang. Thus, for at least the reasons set forth in our Institution Decision, we agree with Patent Owner that Petitioner fails to show that (1) one skilled in the art would have combined the teachings of Chou and Wang or (2) the combined teaching of Chou and Wang renders limitation 1(d) obvious.

e) Conclusion Regarding Claim 1

For the foregoing reasons, we determine that Petitioner has not shown by a preponderance of the evidence that independent claim 1 is unpatentable over the combination of Chou and Wang.

2. *Independent Claims 11, 16, 21, 22, and 26–29 and dependent claims 2, 4, 5, 7, 12, 14, 15, 17, 18, 20, 23, and 25*

Petitioner refers back to its arguments for claim 1 for all limitations of independent claims 11, 16, 21, 22, and 26–29 and, in connection with dependent claims 2, 4, 5, 7, 12, 14, 15, 17, 18, 20, 23, and 25, to its reasons as in Ground 1. *See* Pet. 65. Similarly, Patent Owner relies on the same arguments for claims 11, 16, 21, 22, and 26–29 as for claim 1. *See* PO Resp. 62. Thus, for the reasons discussed in Section III.F.1 above with respect to claim 1, we determine that Petitioner has not shown by a preponderance of the evidence that independent claims 11, 16, 21, 22, and 26–29 are unpatentable over the combined teachings of Chou and Wang. For the same reasons, we also determine Petitioner has not proven by a preponderance of the evidence that dependent claims 2, 4, 5, 7, 12, 14, 15, 17, 18, 20, 23, and 25 are unpatentable the combined teachings of Chou and Wang.

G. Obviousness of Claims 5, 7, and 15 over Chou and Nobuyoshi (Petitioner's Ground 3)

For dependent claims 5, 7, and 15, Petitioner argues with citations to Nobuyoshi and Dr. Reader's testimony that Nobuyoshi teaches each of those claims' additional limitations. Pet. 65–69. Patent Owner does not provide any arguments specifically for any of the challenged dependent claims, other than contending that Nobuyoshi "does not disclose, teach, or suggest the

elements of claim 1 (or the other independent claims)," relying instead on the arguments presented for the independent claims. PO Resp. 62.

Because, as argued by Patent Owner, Petitioner does not apply Nobuyashi for teaching or suggesting the elements of claim 1 and, in particular, limitation [1d], for the reasons discussed in Section III.E.1 above with respect to claim 1, we determine that Petitioner has not shown by a preponderance of the evidence that dependent claims 5, 7, and 15 are unpatentable over the combined teachings of Chou and Nobuyoshi.

IV. CONCLUSION

For the foregoing reasons, we conclude that Petitioner has not demonstrated by a preponderance of the evidence that any of claims 1, 2, 4, 5, 7, 11, 12, 14–18, 20–23, and 25–29 of the '101 patent are unpatentable on the grounds presented.

In summary:

Claim(s)	35 U.S.C. §	Reference(s)/ Basis	Claim(s) Shown Unpatentable	Claim(s) Not Shown Unpatentable
1, 2, 4, 5, 7, 11, 12, 14–18, 20–23, 25–29	103(a)	Chou		1, 2, 4, 5, 7, 11, 12, 14–18, 20–23, 25–29
1, 2, 4, 5, 7, 11, 12, 14–18, 20–23, 25–29	103(a)	Chou, Wang		1, 2, 4, 5, 7, 11, 12, 14–18, 20–23, 25–29
5, 7, 15		Chou, Nobuyoshi		5, 7, 15
Overall Outcome				1, 2, 4, 5, 7, 11, 12, 14–18, 20–23, 25–29

IV. ORDER

In consideration of the foregoing, it is hereby:

ORDERED that claims 1, 2, 4, 5, 7, 11, 12, 14–18, 20–23, and 25–29 of the ’101 patent are not determined to be unpatentable; and

FURTHER ORDERED that, because this is a Final Written Decision, parties to the proceeding seeking judicial review of the decision must comply with the notice and service requirements of 37 C.F.R. § 90.2.